REMARKS

Applicant appreciates the Examiner's withdrawal of the objection to the specification, of the rejections under 35 USC § 101 and of the rejections (partially) under 35 USC § 112.. Applicant has further amended the claims to change all occurrences of "Java" to –JAVA–. The sole remaining issue is the Examiner's use of Cyran for an alleged disclosure of optimizing the runtime environment.

The Examiner characterizes the Cyran patent's disclosure of adding class attribute information to the runtime environment as something "which optimizes the runtime performance." While it is not completely clear, the Examiner appears to be alleging that the attributes in the extended class file 14 are used to optimize the runtime environment. However, these attributes "describe the optimization information generated by the code preparation system 12" (col. 3, lines 10-13), which indicate optimizations to be made to the native code generated by the JIT compiler. To the extent the Examiner may be arguing that Cyran's disclosure of modifying the operation of the JIT compiler based on the attributes in the extended class file, Applicant disagrees that such modifying the operation of the JIT compiler is "optimization."

That is, with regard to the attributes in the extended class file 14, as Cyran discloses, these attributes are generated by the code preparation system 12. Thus, for example, these attributes include "optimization information, hints, and/or directions . . . to use in further processing of the intermediate code." Col. 2, lines 52-54. The code preparation system 12 predetermines optimizations that can be made to the native code. These predetermined optimizations are indicated by the attributes in the extended class file 14. Put simply, Cyran merely discloses that processing that would otherwise be performed by the JIT is performed in advance by the code-preparation system.

While preprocessing by the code preparation system does lessen the load on the runtime environment (i.e., the JIT compiler), this is not optimization of the runtime environment. Rather, this is merely a reallocation of resource usage. For example, see the following portion of Cyran's disclosure (col. 2, lines 52-65):

> The optimization information it provided as additional attributes added to class files 14 generated by the code preparation system 12. It is contemplated that the code preparation system 12 operates on a resource rich computing environment, i.e., has enough space and time resources to effectively implement the optimizations. Thus, by further processing in accordance with the optimization information provided by the code preparation system 12, the code interpretive runtime environment is able to execute the intermediate code more efficiently and to manage its own resources more effectively particularly when executing in a limited resource computing environment. Such limited resource computing environments include set-top boxes, digital personal assistants, etc.

6

On the other hand, referring to Applicant's specification, Applicant discloses actual optimization of the runtime environment – e.g., altogether avoiding processing that can be determined to be unnecessary.

Furthermore, executing optimized native code, without more, cannot be considered optimizing the runtime environment. Even the Examiner appears to recognize the distinction between native code optimization and runtime environment optimization, where the Examiner states: "Therefore, Cyan's disclosure teaches both native code optimization and runtime environment optimization."

For example, (referring to page 9, lines 21-24 of Applicant's specification as filed) one such runtime environment optimization includes loading appropriate features for an application based on the optional attribute(s). This is not a mere reallocation of resource usage but, rather, avoids (or minimizes) unnecessary resource usage. This is optimization. Another optimization disclosed by Applicant includes allocating to a particular portion of memory objects that remain active throughout execution of the Java application. (See page 8, lines 19-21 of Applicant's specification, as filed). Again, this is not a reallocation transfer of resource usage but, rather, avoids (or minimizes) unnecessary resource usage (e.g., by having to reload objects that remain active).

While the discussed distinctions are perhaps subtle distinctions, they are nevertheless patentable distinctions. Contrary to the Examiner's contentions, Cyran does not disclose optimization of the runtime environment.

Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

Alan S. Hodes
Reg. No. 38,185

P.O. Box 70250
Oakland, CA 94612-0250

7.